

CLAIM AMENDMENTS

This listing of claims will replace all prior versions, and listings, of claims in the application.

1 1. (Previously Presented) A data processing system for executing a program
2 having branch instructions therein, each branch instruction specifying a target
3 address in said program defining an instruction that is to be executed if that
4 branch instruction causes said program to branch, said data processing system
5 comprising:

6 a plurality of processing sections, each processing section comprising:

7 a local memory for storing instruction sequences from said
8 program that are to be executed by that processing section, said
9 instruction sequences comprising instructions of different lengths;

10 a function unit for executing instructions stored in said local
11 memory; and

12 a pointer containing a value defining the next instruction in said
13 local memory to be executed by said function unit, wherein each
14 processing section executes part of said program;

15 each function unit executes instructions according to machine
16 cycles, each function unit executing one instruction per machine cycle;
17 and

18 said pointers in each of said processing sections are reset to a
19 new value determined by said target address of one of said branch
20 instructions when a function unit branches in response to that branch
21 instruction.

1 2. (Previously Presented) The data processing system of claim 1 further
2 comprising a memory for determining said new value of said pointers, said
3 memory storing a mapping for each target address in said program specifying
4 one of said pointer values for each of said pointers corresponding to that target
5 address.

1 3. (Previously Presented) The data processing system of claim 1 wherein said
2 program is divided into super instructions, each super instruction comprising a
3 linear block of code that can only be entered at a starting address
4 corresponding to said block of code and each block of code having one or
5 more branch instructions, wherein said target address of at least one of said
6 branch instructions corresponds to a starting address of a super instruction in
7 said program.

1 4. (Previously Presented) The data processing system of claim 1 wherein at
2 least one of said instruction sequences comprises at least one no op
3 instruction.

1 5. (Original) The data processing system of claim 1 wherein said local
2 memory of one of said processing sections comprises a cache memory.

1 6. (Previously Presented) The data processing system of claim 3 further
2 comprising at least one processing section that remains idle for the duration of
3 a super instruction.

1 7. (Previously Presented) The data processing system of claim 3 wherein a
2 super instruction comprises a tuple for each processing section that is not idle
3 for the duration of said super instruction and wherein each tuple identifies a
4 function unit on which execution occurs and a number of memory words
5 needed to represent corresponding operations to be executed on said function
6 unit.

1 8. (Previously Presented) The data processing system of claim 3 wherein a
2 super instruction indicates a total number of machine cycles for said super
3 instruction.

1 9. (Previously Presented) The data processing system of claim 3 wherein said
2 local memories are loaded at a start of said program.

1 10. (Previously Presented) The data processing system of claim 3 wherein
2 said local memories are loaded at the time of a branch to a super instruction.

1 11. (Previously Presented) The data processing system of claim 3 wherein a
2 fall-through super instruction is executed when a super instruction executes
3 without branching.

1 12. (Currently Amended) A data processing system for executing a super
2 instruction according to machine cycles, said super instruction comprising a
3 linear block of code including instruction sequences to be executed by each of
4 a plurality of processing sections and one or more branch instructions, one
5 instruction for each machine cycle, the data processing system comprising:
6 a plurality of processing sections, each processing section comprising:
7 a local memory for storing instruction sequences of the super
8 instruction that are to be executed by that processing section;
9 a function unit for executing instructions stored in said local
10 memory according to machine cycles, each function unit executing one
11 instruction per machine cycle; and
12 a pointer containing a value defining the next instruction in said
13 local memory to be executed by said function unit, the pointers in each
14 of said processing sections being reset to a new value determined by a
15 target address of one of said branch instructions when a function unit
16 branches in response to that branch instruction.

1 13. (Currently Amended) The data processing system of claim 12 wherein
2 said linear block of code of said super instruction can only be entered at a
3 starting address ~~and includes one or more branch instructions.~~

1 14. (Cancelled)

1 15. (Currently Amended) The data processing system of claim 12 ~~[[13]]~~
2 wherein said target address of at least one of said branch instructions
3 corresponds to a starting address of another super instruction ~~in said program.~~

1 16. (Previously Presented) The data processing system of claim 14 further
2 comprising a memory for determining said new value of said pointers, said
3 memory storing a mapping for each target address specifying one of said
4 pointer values for each of said pointers corresponding to that target address.

1 17. (Previously Presented) The data processing system of claim 12 wherein
2 at least one of said instruction sequences comprises at least one no op
3 instruction.

1 18. (Previously Presented) The data processing system of claim 12 said
2 instruction sequences comprise instructions of different lengths.

1 19. (Previously Presented) The data processing system of claim 12 further
2 comprising at least one processing section that remains idle for the duration of
3 the super instruction.

1 20. (Previously Presented) The data processing system of claim 12 wherein
2 said super instruction comprises a tuple for each processing section that is not
3 idle for the duration of said super instruction and wherein each tuple identifies
4 a function unit on which execution occurs and a number of memory words
5 needed to represent corresponding operations to be executed on said function
6 unit.

1 21. (Previously Presented) The data processing system of claim 12 wherein
2 said super instruction indicates a total number of machine cycles for said super
3 instruction.

1 22. (Previously Presented) The data processing system of claim 12 wherein
2 said local memories are loaded at a start of a program that includes said super
3 instruction.

1 23. (Previously Presented) The data processing system of claim 12 wherein
2 said local memories are loaded at the time of a branch to said super
3 instruction.

1 24. (Previously Presented) The data processing system of claim 12 wherein a
2 fall-through super instruction is executed when said super instruction executes
3 without branching.